DOCUMENT RESUME

ED 336 061 IR 015 058

AUTHOR Gunn, Tronie C.
TITLE Mental Models of Adult Logo Students: The Concept of Variable.
PUB DATE Apr 91
NOTE 28p.; Paper presented at the Annual Meeting of the American Educational Research Association (Chicago, IL, April 3-7. 1991).
PUB TYPE Reports - Research/Technical (143) -- Speeches/Conference Papers (150)

EDRS PRICE MF01/PC02 Plus Postage.
DESCRIPTORS *Adult Learning; Demography; Higher Education; Inservice Teacher Education; Microcomputers; Models; Preservice Teacher Education; *Programing
IDENTIFIERS *LOGO Programing Language; *Mental Models

ABSTRACT

This study investigated the mental models acquired by adults who were novice Logo learners with particular attention given to their concept of variable. The study focused on two questions: (1) What mental models are apparently employed for the prediction of Logo events by adult students who are able to write and edit simple Logo procedures? and (2) Are there demographic or experience factors that are consistently correlated with features of the mental models? Useful, less useful, and counter-productive mental models of Logo variables were identified, with less useful models receiving the most support from analysis of subjects' responses, possibly developed by analogy with Logo procedures and mathematical variables. Categorizations based on major field and anxiety level offered clear differences in the models supported. Mathematics and science majors and low anxiety subjects were more likely to support each of the useful models and less likely to support each of the counter-productive models than subjects with other majors and high anxiety subjects. (11 references) (Author/DB)

**********
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
**********

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

☐ This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.



ED336061

# Mental Models of Adult Logo Students:

# The Concept of Variable

Tronie C. Gunn

University of Houston

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

Tronie Gunn

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

Paper Presented at the Annual Meeting of the

American Educational Research Association

April 3 - 7, 1991, Chicago, Illinois

IR015058



BEST COPY AVAILABLE

## Abstract

This study investigated the mental models, particularly regarding their concept of variable, acquired by adults who were novice Logo learners. Specifically, the study focused on two questions: (a) What mental models are apparently employed for prediction of Logo events by adult students who are able to write and edit simple Logo procedures? (b) Are there demographic or experience factors which are consistently correlated with features of the mental models?

Although subjects rarely gave responses which specifically identified their mental models of Logo variables, several possible models were apparently being used: (a) useful models, such as the "container" model and the "slate" model; (b) less useful models, such as the "procedures" model and the "names" model; and (c) counter-productive models, such as the "smart computer", "black box", and "use everything" models. The models that received the most support from analysis of subjects' responses were the less useful models, perhaps developed by analogy with Logo procedures and mathematical variables. Categorizations based on major field and anxiety level offered clear differences in the models supported. Mathematics and science majors and low anxiety subjects were more likely to support each of the useful models and less likely to support each of the counter-productive models than subjects with other majors and high anxiety subjects.

# Mental Models of Adult Logo Students: The Concept of Variable

"Mental models are the mechanisms whereby humans are able to generate descriptions of system purpose and form, explanations of system functioning and observed system states, and predictions of future system states." (Rouse & Morris, 1985, p. 7) Although development of complex concepts may begin while students are in an instructional setting, their mental models of the systems involved would likely influence the development of those concepts after the students are removed from the guidance of an instructor. As a result, it seems important to investigate the mental models of complex concepts acquired during instruction to assess whether or not they will tend to be productive or counter-productive in the development of mature concepts.

The concept of variable is both inconstant and elusive. It is hard to describe (Schoenfeld & Arcavi, 1988) and hard to learn (Rosnick, 1981). A survey of definitions of variable reveals that it means different things in different contexts. Understanding the mathematical concept of variable provides the basis for the transition from arithmetic to algebra and is necessary for the meaningful use of all advanced mathematics. In the same way, understanding the computing concept of variable is the key to the production of computer programs that have the desirable property of implementing general algorithms that can be applied to a variety of data.

In many uses of programming variables, the mathematical concepts of variables and equality would be sufficient to determine the result of a programming instruction. For example, for assignments that only depend on constant values and/or other variables, a unique value can be associated with the name of the assigned variable. However, the understanding of the incrementation/ decrementation assignment makes it necessary to consider a temporal aspect: the value of the variable after the assignment is replaced by the value of the variable before the assignment, plus or minus some value. The student must designate by the same name both the preceding value and the present value (which depends on the former), and has to process the assignment as an asymmetric relation. It seems likely, therefore, that understanding this type of assignment will be extremely difficult for novices who reason from the mathematical concept of variables.

Subjects of all ages have demonstrated misconceptions related to the concept of programming variables. Some of the most frequently documented misconceptions have been an assumed relationship between the variable's name and its value or function in the program and misunderstandings related to the passing of parameters and the scope of variables between and within procedures (e.g., DuBoulay & O'Shea, 1981; Kurland & Pea, 1985; Lee & Lehrer, 1987; Nickerson, Perkins, & Smith, 1985).

Students may not articulate a clear picture of the mental models they use to make their predictions and explanations. However, based on the patterns of correct responses, errors, and misconceptions, it is possible to infer broad characterizations of the mental models that may have been used. No claim is made that these models are the only ones that could produce the observed results.

Some potential mental models relate directly to the subjects' concepts of computing variables. These models include the "container" model, the "slate" model, and the "procedures" model. Each of these is explained in more detail below. Although there are potential problems with each of these models, they are generally productive models in appropriate situations. Another model of computing variables, based largely on the mathematical concept of variables, might also be used. It is described below as the "names" model. Unfortunately, the differences between the mathematical concept and the computing concept of variables makes this model misleading in some very simple examples.

**The "Container" model.** In this model, subjects view variables as named containers for values. This type of model is one of the most commonly suggested models for variables in textbooks and typical programming instruction (e.g., Fox & Waite, 1981; Luehrmann & Peckham, 1983). Programming instruction typically stresses that these containers (variables) can only hold a single value at a time. However, students may imagine that the containers (variables) can hold many values at once, just as the real-world containers with which they are familiar can

hold many things at one time (DuBoulay, 1989). Although imagining potentially empty containers in a situation such as Logo programming that does not allow empty variables to exist is unlikely to cause difficulties in making predictions or explanations, it is another possible shortcoming of this model.

An important potential shortcoming of this model becomes obvious when the possibility of both local and global variables is considered. To account for the possibility of both local and global variables, it is necessary to allow for multiple sets of containers to exist at the same time. The containers within each set would be required to have unique names, but containers in two different sets could have identical names, because the set to which the container belonged would be a distinguishing identifier. There also would be one set designated as the "global" set, which would become, in effect, joint property of all the "local" sets of containers and would be referenced if no "local" container had an identical name.

**The "Slate" model.** In this model, computer memory is viewed as containing an erasable slate on which the names and current values of variables can be written. A memory reference copies the contents from the slate, while an assignment writes a new value, after erasing the previous one, if any. Subprocedure calls can be correctly viewed as not simply passing control to a new set of commands (as in BASIC), but also making a set of variable assignments based on the current values of specified variables (or constants). Subjects with this model would have no difficulty with multiple assignments - the slate can only

have one value (the latest assigned) for a given variable. However, to successfully deal with the complexity of local and global variables, this model might also prove cumbersome.

**The "Procedures" model.** Subjects could also overgeneralize their models of procedures in Logo to use them as models for variables. This could result in the assumption that incrementation/decrementation assignments would create new copies of the variables named in a manner similar to recursion of procedures. As there is no command-based equivalent of in-place replacement of a procedure in Logo, generating an explanation of this type of command could be difficult from this model. Once a procedure is defined, the definition is stored ("remembered") and is later invoked by naming it. This might lead to the expectation that assignment equations are remembered and can reproduce the specified action later by name. Object Logo does not allow the user to define a procedure with the same name as one already defined in the workspace; an error message results. Subjects using this model could expect that repeated assignments would also result in an error message or that an assignment to a variable called W might cause a problem because of the already defined procedure named W.

**The "Names" model.** Subjects with this model would seem to view variables as location-less abstractions, similar to mathematical variables. These variables can have many values, but only one at a time (for a given exercise). However, the value is linked to the name and the name is globally known (at least

for the duration of that exercise). This model seems to be basically parallel and static. A temporal dimension is definitely missing.

This model causes severe problems in encounters with incrementation/ decrementation assignments. It is not reasonable to consider that C can in any way be or become C+1. Though Logo uses a command (MAKE) that suggests the temporal nature of the assignment, instead of the potentially more confusing equal sign used in some languages, subjects using this model could still be confused.

Another error that may have come from this model involves difficulties with repeated assignments. Making two incompatible assignments (such as making TIMES both 15 and 25) is incredibly confusing when processed through the names model.

This model seems unlikely to evolve into a more adequate model without knowledgeable intervention. Subjects have a wide range of examples in mathematics upon which to base this model. The limited exposure to programming in an overview course is unlikely to offer sufficient examples to discard this model, which worked successfully in other contexts. The instructor must clearly present and explain examples for which this model is insufficient and misleading.

**Other models.** Several mental models that could be used do not relate directly to variables. Instead, they relate to an overall view of computers and problem-solving. These include (a) the "smart computer" model, in which the computer or the currently executing program is expected to possess intelligence,

understanding, and will; (b) the "black box" model, in which the functioning of the computer is considered mysterious and not easily comprehensible; and (c) the "use everything" model, in which problem solving is viewed as simply combining as many as possible of the apparent facts into the "right" form. These models are highly counterproductive, directing the attention to features that are only coincidentally related to the actual results.

## The Study

An exploratory study, conducted in the spring of 1990, explored the mental models acquired by preservice and inservice teachers in the context of computer literacy Logo instruction. As part of the study, the subjects' concepts of computing variables was explored through prediction and explanation exercises. Specifically, the study focused on two questions: (a) What mental models are apparently employed for prediction of Logo events by adult students who are able to write and edit simple Logo procedures? (b) Are there demographic or experience factors which are consistently correlated with features of the mental models?

## Methodology

Subjects were 13 student volunteers enrolled in sections of the undergraduate and graduate computer literacy for teachers courses offered at a major Southwestern university during spring of 1990. They participated in the routine course introduction to the Logo computer language with their class sections, completing the same assignments and activities as other students enrolled in the courses. To insure comparability between the Logo experiences of the graduate and undergraduate students, the instructor for the graduate course presented the lecture on variables for both courses. The student volunteers were asked to complete a more detailed experience and demographic questionnaire than that routinely used in the courses. Based on that information, subjects were selected for diversity along several dimensions: (a) classification, (b) gender, (c) teaching experience, (d) programming experience, (e) major field, (f) self-reported computer anxiety, and (g) age. These demographic results were used as categorical variables in the analysis of results.

Subjects were then asked to complete three additional tasks: (a) a booklet in which they were asked to write down in ordinary English "what happens inside the computer when ..." specific examples of Logo MAKE, PRINT, and REPEAT commands are executed (similar to the booklet developed for BASIC by Bayman & Mayer, 1983), (b) a videotaped interview in which they were questioned about

their understanding of "variable" as a computing concept, and (c) a series of brief Logo procedures for which they were asked to predict and explain the results of the procedures. This paper will briefly report the results of the first and second tasks, while focusing on the third.

Subjects' descriptions of the executions of the specified commands were broken down into transactions (a specified operation applied to a specified object at a specified location). The most common correct and incorrect transactions for each command were summarized and compared for demographic and experience subgroupings. Common responses to the questions asked in the variables interview were also summarized. The results of both tasks were reviewed for evidence of the use of specific models.

Subjects' predictions and explanations for the Logo procedures were reviewed and recurring errors and evidence of misconceptions were tallied. Based on these tallies and the characterizations of the possible models, support for each of the proposed models was identified. Results were compared for demographic and experience subgroupings.

**Results: The Logo Booklet**

Three categories of language statements that were studied included direct assignments (using MAKE), screen display (using PRINT), and repetition (using

REPEAT). The subjects tended to verbalize conceptions of the statements that either failed to include the main idea or that included outright misconceptions for all three categories of statements. For example, the location in which a value obtained by incrementation would be stored (replacing the variable's previous value) was not clear and subjects commonly indicated that all assignment statements also produced some form of screen display. Although all subjects had demonstrated "hands-on" competence in using Logo, this experience was not sufficient for the mature understanding of programming.

Some of the common responses, especially those for the assignment statements (summarized in Table 1), might be related to the mental model of variables used by each subject. For example, subjects who indicated that equations were recorded or remembered or that a new variable was created by an incrementation/decrementation assignment might be using a "procedures" model. This type of error (and, hence, suspected use of the "procedures" model) was much more likely for teachers than non-teachers (see Table 2).

**Results: The "Variable" Interview**

Subjects generally exhibited relatively incomplete concepts of variables. Several subjects demonstrated some mixing of the ideas of variables and procedures in Logo. The subjects generally presented naive but reasonable

descriptions of variables as placeholders or substitutes (similar to the mathematical concept of variable). They identified some defining characteristics of variables but omitted others. These results are summarized in Table 3. It is the lack of inclusion of location as a critical, required component of a computing variable that may have been responsible for many of the observed misconceptions. This omission suggests the possible use of the "names" model.

Although most suggested that there was a command that could be used to assign a value to a variable, only a few (16.7%) were able to correctly identify the MAKE command. The same number of subjects (16.7%) incorrectly assigned that function to the TO command, which is used in the definition of procedures (possibly as a result of use of the "procedures" model).

**Results: The Logo Prediction/Explanation Examples**

The most common errors and misconceptions observed in the prediction/explanation portion of the study are presented in Table 4. Many of these errors were similar to those observed in the Logo booklet transaction lists. These include a variety of difficulties with assignments (especially incrementation/decrementation) regarding their use in repetition. Some of these errors might be related to the mental models used (For example, both the "names"

model and the "procedures" model would make it difficult to understand incrementation/decrementation).

Of the errors not obviously related to the transaction lists, perhaps the most common were those that attribute to the computer human-like will or understanding. Subjects who suggested that the actions of the computer were not within programming control or that the computer could understand the names of procedures and/or variables and act accordingly evidence counterproductive mental models (perhaps the "black box" or "smart computer" models characterized earlier). High anxiety subjects and subjects whose major field was not mathematics or science were the most likely to make these errors. Many of these errors could also indicate that the subjects actually had no mental model of computing variables or that they were utilizing the "use everything" model of problem solving.

Although some of the common errors probably do not serve to differentiate between mental models used, other errors might reasonably be related to the use of specific mental models. For example, allowing two variable values to coexist would be most compatible with the containers model (without sufficient limitations on the number of concurrent values allowed). Users of the "procedures" model might be expected to have difficulties with the simultaneous existence of a W procedure and a W variable.

Table 5 presents the proposed models and the support they received by the responses to the prediction/explanation tasks. It is interesting to note that the most supported model is the "procedures" model, followed closely by the "names" model. There was also fairly strong support for an explanation of no model or the use of "bad" models, such as the "use everything" and "smart computer" models. Undergraduate students' responses gave significantly more support to the "names" model than did graduate students' responses. Teachers' and non-programmers' responses were more likely to support the use of no model and the "use everything" model than non-teachers' and programmers' responses. Responses for the categories based on self-reported anxiety level and major field, however, were significantly different in support of every model. Mathematics and science majors and low anxiety subjects were significantly more likely to support the "better" models ("slate," "container," "procedures," and "names") and significantly less likely to support no model or "worse" models ("smart computer," "black box," and "use everything") than non-mathematics and science majors and high anxiety subjects. These results are presented in Tables 6, 7, and 8.

## Conclusions

Subjects seldom articulated a clear picture of the mental models they may have been using to make their predictions and explanations. Subjects with "smart

computer" and "black box" models could be expected to have limited future development of their concepts of variables without further instruction. Their mental models of computing lead them to feel that they cannot control or understand what happens, so they make little effort to do so.

Both the "container" and "slate" models generally supported good predictions about the functioning of Logo variables. One subject who consistently referred to an analogy of variables with boxes obviously used a containers model. This model was clearly useful for her. Other subjects may have also used this model. The slate model was not directly observed as the only possible explanation of the predictions of any subjects, but may have been the underlying model of many successful subjects. Instructors should be encouraged to teach these models of variables, with their limitations. Experience with the Logo environment in particular and with computing in general might sufficiently identify and clarify limitations of each model (without further formal instruction) so a mature concept of programming variable can result (perhaps without the component of scope, unless specific knowledgeable intervention occurs).

The most commonly supported model was not either of these "most useful" models. It was, instead, a model based on overgeneralization of Logo procedures as models for Logo variables. This "procedures" model received clear support in all three tasks and it was clearly useful in some situations. Subjects with limited experience with Logo variables may have been able to make successful predictions

about variables by applying this analogy. However, the limitations of the analogy were not as clear as its application. Subjects encountered situations (such as incrementation/decrementation assignments) for which procedures offered them either no guidance or misleading guidance in making predictions. And when these predictions were demonstrated to be inaccurate, the subjects often had no better means of explaining the error than by relying on either the "smart computer" or "black box" models.

Another commonly supported model was the "names" model, based on the mathematical concept of variables. Subjects utilizing the "names" model seem to be doomed to experience confusion and frustration. Subjects can readily construct this mental model based on years of exposure to it through mathematics. Unfortunately, the fundamental natures of mathematical variables and computing variables are different! Mathematical variables are abstractions - they can only be identified by their names. But computing variables are more than just a name! Computing variables have a location - the name is really only a convenience for the programmer. Computing variables with the same names can assume different functions within the context of the same program and computing variables with different names can communicate values from one part of a program to another. Care should be taken to eliminate this model from the repertoire of all computing students, but especially those who plan to become teachers and who might pass this model on to other students.

BIBLIOGRAPHY

Bayman, P., & Mayer, R. E. (1983). A diagnosis of beginning programmers' misconceptions of BASIC programming statements. Communications of the ACM, 26, 677-679.

DuBoulay, B. (1989). Difficulties of learning to program. In E. Soloway & J. C. Spohrer (Eds)., Studying the novice programmer (pp. 283-299). Hillsdale, NJ: Lawrence Erlbaum.

DuBoulay, B., & O'Shea, T. (1981). Teaching novices programming. In M. J. Coombs & J. L. Alty (Eds.), Computing skills and the user interface (pp. 147-200). New York: Academic Press.

Fox, D., & Waite, M. (1981) Pascal primer. Indianapolis: The Waite Group, Howard W. Sims.

Kurland, D. M., & Pea, R. D. (1985). Children's mental models of recursive Logo programs. Journal of Educational Computing Research, 1, 235-244.

Lee, O., & Lehrer, R. (1987, April). Conjectures concerning the origins of misconceptions in LOGO. Paper presented at the Annual Meeting of the American Educational Research Association, Washington, DC.

Luehrmann, A., & Peckham, H. (1983). Computer literacy: A hands-on approach. Mission Hills, CA: Glencoe/McGraw-Hill.

Nickerson, R. S., Perkins, D. N., & Smith, E. E. (1985). The teaching of thinking. Hillsdale, NJ: Lawrence Erlbaum.

Rosnick, P. (1981). Some misconceptions concerning the concept of variable. The Mathematics Teacher, 74, 418-420.

Rouse, W. B., & Morris, N. M. (1985, May). On looking into the black box: Prospects and limits in the search for mental models (Report No. ONR-TR-85-2). Arlington, VA: Office of Naval Research. (ERIC Document Reproduction Service No. ED 268 131)

Schoenfeld, A. H., & Arcavi, A. (1988). On the meaning of variable. The Mathematics Teacher, 81, 420-42.

Table 1. Percentage of Subjects Who Produced Listed Transactions for Selected Prestatements

| Prestatement/Transaction | % Subjects |
|---|---|
| Assign: Increment/Decrement | |
| MAKE "B :B + 1 MAKE "C :C - 1 | |
| ! write the obtained value in specified memory space. | 65 |
| print the obtained value. | 27 |
| record/remember the equation. | 23 |
| create a "new" variable. | 12 |
| Assign: Counter set to number | |
| MAKE "D 0 | |
| ! write zero in memory space D | 54 |
| print the value | 23 |
| Assign: Mixed formula | |
| MAKE "A :B + 1 | |
| ! write the obtained value in memory space A | 77 |
| print the obtained value | 23 |
| record/remember the equation | 23 |

Table 2. Selected Transactions - Subgrouping by Teaching Experience

| Prestatement/Transaction | Teaching Experience | No Teaching Experience | Chi Square |
|---|---|---|---|
| Assign: Counter set to number<br>MAKE "D 0 | | | |
| ! write 0 in memory space D | 75.0% | 44.4% | 1.04 |
| Assign: Increment/Decrement<br>MAKE "B :B + 1<br>MAKE "C :C - 1 | | | |
| ! write value in mem space | 62.5% | 66.7% | 0.04 |
| remember the equation | 50.0% | 11.1% | 4.72 * |
| create a "new" variable | 25.0% | 5.6% | 2.05 |
| Assign: Mixed Formula<br>MAKE "A :B + 1 | | | |
| ! write value in mem space A | 75.0% | 77.8% | 0.01 |
| remember the equation | 75.0% | 0.0% | 8.78 * |

* $p < .05$, $df = 2$.

Table 3. Components of the Concept of Variable Observed in "Variables" Interview

| | |
|---|---|
| Variables must have: | |
| * a name | 83.3% |
| * a value | 58.3% |
| * a location | 33.3% |
| * a scope | 0.0% |
| Variable characteristics: | |
| * name can be changed | 83.3% |
| * only 1 name at a time | 75.0% |
| * value can be changed | 100.0% |
| * only 1 value at a time | 91.7% |
| Value assignment by: | |
| * MAKE command | 16.7% |
| * TO command | 16.7% |
| * other or unknown command | 58.3% |

Table 4. Prediction/Explanation Error Summary

| Number | Error | Percent | Possible |
|---|---|---|---|
| 1. | REPEAT/Increment Confusion | 41.2% | 17 |
| 2. | Variable with 2 values at once | 19.0% | 63 |
| 3. | MAKE => printing | 7.9% | 76 |
| 4. | W as variable/procedure? | 12.5% | 24 |
| 5. | Order of MAKES => function | 11.8% | 76 |
| 6. | Can't predict in advance | 15.8% | 76 |
| 7. | Name of procedure => shape | 12.9% | 62 |
| 8. | Name of variable => function | 28.8% | 52 |
| 9. | Pick function based on #s | 10.5% | 76 |
| 10. | "Smart Computer" average (#6-9) | 17.0% | |

Table 5. Models Supported Through Prediction/Explanation Responses

| Model | Percent |
|---|---|
| SLATE Model | 64.10% |
| CONTAINER Model | 66.67% |
| PROCEDURES Model | 84.62% |
| NAMES Model | 82.05% |
| SMART COMPUTER Model | 39.74% |
| BLACK BOX Model | 24.36% |
| USE EVERYTHING Model | 44.87% |
| No Model | 51.28% |

Table 6. Models Differentially Supported

| Model | Undergraduate | Graduate | Chi Square |
|---|---|---|---|
| NAMES Model | 90.48% | 72.22% | 4.39 * |

* $p < .05$, $df = 2$.

| Model | Teaching Exp. | No Teaching | Chi Square |
|---|---|---|---|
| USE EVERYTHING Model | 66.67% | 35.19% | 6.66 * |
| No Model | 66.67% | 44.44% | 3.28 |

* $p < .05$, $df = 2$.

| Model | Programming | No Prior Programming | Chi Square |
|---|---|---|---|
| USE EVERYTHING Model | 33.33% | 58.33% | 4.90 * |
| No Model | 40.48% | 63.89% | 4.25 * |

* $p < .05$, $df = 2$.

Table 7. Models Supported by Major Field

| Model | Science/ Mathematics | Non-Science Mathematics | Chi Square |
|---|---|---|---|
| SLATE Model | 77.78% | 52.38% | 5.43 * |
| CONTAINER Model | 83.33% | 52.38% | 8.36 * |
| PROCEDURES Model | 100.00% | 71.43% | 12.16 * |
| NAMES Model | 97.22% | 69.05% | 10.45 * |
| SMART COMPUTER Model | 19.44% | 57.14% | 11.50 * |
| BLACK BOX Model | 8.33% | 38.10% | 9.32 * |
| USE EVERYTHING Model | 19.44% | 66.67% | 17.47 * |
| No Model | 22.22% | 76.19% | 22.60 * |

* $p < .05$, $df = 2$.

Table 8. Models Supported By Anxiety Level

| Model | Low Anxiety | High Anxiety | Chi Square |
|---|---|---|---|
| SLATE Model | 69.70% | 33.33% | 5.83 * |
| CONTAINER Model | 72.73% | 33.33% | 7.09 * |
| PROCEDURES Model | 90.91% | 50.00% | 13.05 * |
| NAMES Model | 87.88% | 50.00% | 9.89 * |
| SMART COMPUTER Model | 31.82% | 83.33% | 11.25 * |
| BLACK BOX Model | 18.18% | 58.33% | 8.88 * |
| USE EVERYTHING Model | 34.85% | 100.00% | 17.42 * |
| No Model | 42.42% | 100.00% | 13.47 * |

* $p < .05$, $df = 2$.